

# **Plant Identification from Leaves using Pattern Recognition Techniques**

**Ishank Kumar Rawat**



Department of Electronics & Communication Engineering  
**National Institute of Technology Rourkela**

# **Plant Identification from Leaves using Pattern Recognition Techniques**

*Thesis submitted in partial fulfillment*

*of the requirements of the degree of*

***Master of Technology***

*in*

***Electronics & Communication Engineering***

*by*

***Ishank Kumar Rawat***

(Roll Number: 214EC6210)

*based on research carried out*

*under the supervision of*

***Prof. Ajit Kumar Sahoo***



May, 2016

Department of Electronics & Communication Engineering  
**National Institute of Technology Rourkela**



Department of Electronics & Communication Engineering  
**National Institute of Technology Rourkela**

---

May 26, 2016

## **Certificate of Examination**

Roll Number: *214EC6210*

Name: *Ishank Kumar Rawat*

Title of Dissertation: *Plant Identification from Leaves using Pattern Recognition Techniques*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Master of Technology* in *Electronics & Communication Engineering* at *National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness, and originality of the work.

---

Ajit Kumar Sahoo  
Principal Supervisor



Department of Electronics & Communication Engineering  
**National Institute of Technology Rourkela**

---

**Prof. Ajit Kumar Sahoo**

Professor

May 26, 2016

## **Supervisor's Certificate**

This is to certify that the work presented in the dissertation entitled *Plant Identification from Leaves using Pattern Recognition Techniques* submitted by *Ishank Kumar Rawat*, Roll Number 214EC6210, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Master of Technology in Electronics & Communication Engineering*. Neither this thesis nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

---

Ajit Kumar Sahoo

# **Dedication**

***To My Family & Friends...***

# Declaration of Originality

I, *Ishank Kumar Rawat* , Roll Number *214EC6210* hereby declare that this dissertation entitled *Plant Identification from Leaves using Pattern Recognition Techniques* presents my original work carried out as a postgraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections “Reference” or “Bibliography”. I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

May 26, 2016  
NIT Rourkela

*Ishank Kumar Rawat*

# Acknowledgement

This work would not have been possible without the help of many people who constantly supported me for which I am grateful to them. With profound respects and significant admiration, I avail this chance to express my profound feeling of gratitude and obligation to **Prof. Ajit Kumar Sahoo**, Department of Electronics and Communication Engineering, NIT Rourkela for their significant guidance and support. I am profoundly obliged for the profitable discussions at every phase of the project. I consider it as my good fortune to have got an opportunity to work with such wonderful personalities.

I want to express my sincere gratitude to Prof. K.K. Mahapatra, HOD, Department of Electronics and Communication Engineering for creating an environment for study and research. Sincere thanks to Prof. S. Meher, Prof. U.K. Sahoo, Prof. Lakshi Prosad Roy, Prof. S. Ari. and Prof. Manish Okade for helping me learn and for their constant encouragement. I want to thank all faculty members and staff of the Department of Electronics and Communication Engineering, NIT Rourkela for their generous help.

I would like to make special mention of the selfless support and guidance I received from my classmates for their endless companionship and help. They made my stay at NIT, Rourkela enjoyable.

I am especially indebted to my parents, brother for their constant love and guidance in every step of my life.

# Abstract

Medicinal plants have been used throughout the human history. Ayurveda is one of the oldest medicine system, which is even recognized in the modern medical society, uses plants for the preparation of medicines. There are thousands of species of plants used in the preparation of medicines. The difficulty lies in the identification of plant species. An individual with deep knowledge of plants can only differentiate between these species. This makes leaf identification very difficult. A reference guide to plants identification may ease up the problems. This is where nature needs engineering. In this work, a system is being developed which helps in the identification of the plants based on the leaf. This system takes input as a leaf image and outputs the name of the species and other relevant details which are stored in the database.

The system is designed using the technique of image identification using *pattern recognition*. The approach of shape and texture identification, both are combined for designing such a system. The segmentation of the images was done using the techniques of ***graph-cuts***. The descriptor used for shape identification was ***Shape Context*** and textures were described using ***Local Binary Patterns***. The classification was done using feed forward ***Multi-Layered Perceptron (MLP)*** neural network with *backpropagation training* algorithm. The system was tested of certain class of leaves and the performance of the system is compared with an existing system.

***Keywords:*** *Graph Cuts, Shape Context, Local Binary Patterns, Multi-Layered Perceptron (MLP).*



# Table of Contents

<i>Chapter Name</i>	<i>Page No.</i>
Chapter 1: Introduction.....	- 1 -
1.1 Motivation .....	- 2 -
1.2 Literature Review .....	- 3 -
1.3 Problem Statement.....	- 4 -
1.4 Summary of Chapters .....	- 4 -
Chapter 2: Image Segmentation .....	- 5 -
2.1 Introduction .....	- 6 -
2.2 Image Segmentation .....	- 6 -
2.2.1 Otsu's method of image segmentation .....	- 6 -
2.2.2 k-Means Method of Image Segmentation .....	- 8 -
2.2.3 Graph Cuts .....	- 9 -
Chapter 3: Computer Vision.....	- 13 -
3.1 Introduction .....	- 14 -
3.2 Shape Descriptors .....	- 15 -
3.2.1 Shape Context.....	- 15 -
3.3 Texture Descriptors .....	- 18 -
3.3.1 Local Binary Patterns .....	- 19 -
3.4 Application to Project .....	- 21 -
Chapter 4: Pattern Recognition.....	- 22 -
4.1 Introduction .....	- 23 -
4.2 Neural Networks.....	- 24 -

4.3 Application to Project .....	31 -
Chapter 5: Plant Identification Framework .....	32 -
5.1 Proposed Model .....	33 -
5.2 Experiment.....	34 -
5.3 Results .....	35 -
5.3.1 Image Preprocessing and Segmentation .....	35 -
5.3.2 Shape Matching .....	37 -
5.3.3 Texture Matching .....	38 -
5.4 Classification Results .....	39 -
Chapter 6: Conclusion & Future Scope .....	43 -
6.1 Conclusion .....	44 -
6.2 Future Scope .....	44 -
References .....	46 -

# **Chapter 1: Introduction**

## **Introduction**

## **1.1 Motivation**

India is a country a country of diversities. With temperature variations from  $-2^{\circ}\text{C}$  to  $47^{\circ}\text{C}$  with three cropping seasons namely: Rabi, Kharif, and Zaid. Each season has a different amount of rainfall. There are a large number of grown annually in India. Due to varying climate conditions and all such factors, the plants are vulnerable to some very harmful diseases. With knowledge of plants confined to only a few scholars, the common mass finds this tough to understand the plants. This project will help will serve as bridge connecting the knowledge of botanists to common mass, helping them to understand the plants. From gardening to farming, this project would serve them all.

While there are a huge number of applications available over internet and in research paper dealing with identification of plants, none of them are attempting to automate the process of identification. Various apps have already been developed which are based on the database and logs stored. But the drawback of these apps is one has to manually iterate through the entire list available. Attempts have been made to narrow down the process of search using searching based on keys such as: search based on shape entered, or the shape of leaf described by user. Some of the apps are based on the on the services. One can consult the botanist. Because of the direct human involvement, there is more accuracy in such services; the process is time consuming, as one has to wait for the response. Also the cost of implementing such a service based system will be huge. Moreover these applications are online and some are not even standalone. Project can be viewed as an improvisation on the present techniques of identification. This project is a perfect combination of image processing, data mining and pattern recognition.

The project can be assumed much similar to identification of face or fingerprint recognition. But the project is very different from the above stated project. It is so because in the identification of biometrics or face, the broad structure of the face or fingerprints remain same. While in plant recognition, the shape of one class of leaf may vary very less significantly vary from another. Moreover the problem arises when leaves of the same class vary drastically within same class.

## 1.2 Literature Review

Because there is a lot of variation in identification, one cannot simply use just a single method for identification purpose. A similar research in the identification of face and biometrics has been performed in the identification of faces by A. K. Jain et. al. in his paper “*An introduction to biometric recognition*” [3]. In this paper, he proposed the global analysis of feature, such as in face identification, the relative positions of eyes and noses were the key elements. He also gave the quantitative system of measurement to keep a check on accuracy measurement on the system. Another paper by Lin Lin Shen and Li Bai presented the identification based on the General Discriminant analysis and Gabor Wavelets features [4]. The Gabor Wavelets have been extensively used in texture identification. The Gabor filter was first introduced by Nobel laureate *Dennis Gabor*. This filter was used for modelling simple cells in visual cortex of mammal’s brain. This filter was based on the edge detection and has been used in applications such as optical character recognition, facial expression recognition, fingerprint recognition iris recognition, etc. The paper presented by Jyotismita Chaki [5] discusses the use of this Gabor filter in the recognition of leaves. Another paper “*Ap Leaf: An efficient android-based plant leaf identification system*” [1] attempts the automatic segmentation of leaves combined the Gabor Wavelet filter with the other features such as Pyramid histograms of oriented gradients (PHOG) and color information for the automatic segmentation of leaves. This paper “*Plant leaf recognition using texture and shape features with neural classifiers*” [6] combines the shape and texture based approach. The texture extraction is based on the Gabor Filter and Gray Level Co-occurrence Matrix (GLCM) and the shape estimation is based on the curvelet transform and Invariant Moments. In this paper the shape matching technique given in the paper “*Shape Context: A new descriptor for shape matching and object recognition*” [7] is followed. This is one of the most popular techniques of shape identification and matching. For the texture identification, the technique of Local Binary Patterns (LBP) is used. The LBP operator was first introduced by *Ojala et. al.* in the paper “*Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*” [8]. The local binary patterns are excellent texture descriptors. A few application of LBP are in Finger Print recognition [9] and Facial Recognition [10]. The paper “*Plant identification using leaf shapes - A pattern counting approach*” [2] proposes a more advanced method of shape matching by extending the Inner Distance Shape Context (IDSC) [11].

## **1.3 Problem Statement**

The project is addressed to problem of automatic identification of plants based on the leaf image. The aim is to prototype a system that performs the same. The system is designed for identifying a leaf from certain class of image. In the project is discussed the approach to design such a system. The thesis also clearly mentions the techniques which should be used and which should not be used for the identification process.

## **1.4 Summary of Chapters**

In the project the techniques of Rotation Invariant LBP, Shape Context and Graph Cuts are combined and classifications are done using Neural Networks. The project is thoroughly described with every single details of implementation. As already described, the first chapter presents the introduction and literature review. The rest of the thesis is formatted in this order: Second chapter deals with the segmentation of images. Third chapter describes about the techniques of image processing which are used in the project. Fourth chapter deals with the techniques of pattern recognition for the classification of various species of plants. Fifth chapter describes about the problem statement and the proposed framework for the thesis. This chapter includes in details of the entire framework and the results of the performed experiments. Finally, in sixth chapter, conclusion of the thesis is presented.

## **Chapter 2: Image Segmentation**

### **Image Segmentation**

## 2.1 Introduction

Image processing is operations performed on a digital image for the purpose of enhancement, data extraction or some other application. Image enhancement has no formal definition, but vaguely it can be said that Digital image enhancement deals with the alteration of digital image so as enhance the visual context, so that the image is more perceivable. One of the application used in this project is in computer vision. In the computer vision, there is a need for identification of the object of interest from the image itself. Thus it is required to separate out the object of interest from the background. This is called segmenting image. But if the image is itself corrupted by noise, then the segmentation is not perfect and erroneous edges are detected of the object under study which at the end leads to erroneous results. Thus it was observed that the image segmentation is a very important step in identification of objects. And if done wrongly, a chain of error propagates down from the top to the bottom.

## 2.2 Image Segmentation

As described it is very important for the proper functionality of the project, image segmentation should produce minimum errors. To reduce this error, it is required that the background of the image must be in contrast with the foreground of the image. In the project of identification of leaf, it is possible to take images in the lab with a white background, which is in contrast with the greener portions of leaf. One method of segmenting the image is Otsu's [12] method of image global segmentation.

### 2.2.1 Otsu's method of image segmentation

A histogram of an image represents the counting of number of pixels for each intensity level in an image. Otsu's method of image segmentation is based on the thresholding operation. In binary image segmentation, the thresholding operation means, all the pixels belonging to some predefined criteria are assigned one labels and rest of the pixels are assigned different labels. In this way the image is divided into two classes of pixels: the foreground and background. Generally this type of segmentation is used when all the pixels belonging to foreground class are largely similar and are very different to the pixels of background. On observing the histogram of such images, there must



be two modes of histogram clearly distinguishable. Such a similar histogram is shown in figure (2.1)

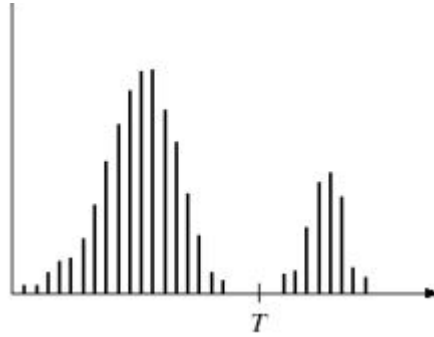


Figure 2.1, Bi-modal histogram with threshold  $T$

As can be clearly seen from figure (2.1) there are two modes in histogram which are clearly separable and one such good choice of choosing threshold is  $T$ . Each mode may represent a separate object. Otsu's method of segmentation is method of finding the optimal value of threshold such that the two classes are clearly separable.

This method is based on selecting such a threshold such that the weighted within class variance is minimized. It can also be seen that upon selecting that threshold, the inter classes variance is also maximized.

Consider an image of  $M \times N$  size. Let intensities in the image ranges from  $0 - L-1$ . The normalized satisfies the condition-

$$\sum_{x=0}^{L-1} p_x = 1, \quad p_x \geq 0 \quad (2.1)$$

Where  $p_x$  is the probability given by  $n_x/MN$ .  $n_x$  is number of pixels corresponding to intensity  $x$ . Let the selected threshold be  $k$  such that  $0 < k < L - 1$ . This threshold divides the image into two classes  $C_1$  and  $C_2$ , where  $C_1$  is the class consisting of pixel intensities ranging from  $[0, k]$  and  $C_2$  ranges from  $[k+1, L-1]$ . Then  $P_1(k)$  is the probability that pixel belongs to class  $C_1$ . Then

$$P_1(k) = \sum_{x=0}^k p_x \quad (2.2)$$

And similarly  $P_2(k)$  is the probability that pixel belongs to class  $C_2$ .

$$P_2(k) = \sum_{x=k+1}^{L-1} p_x = 1 - P_1(k) \quad (2.3)$$

The mean intensity levels for class  $C_1$  and  $C_2$  is given by-

$$m_1(k) = \frac{1}{P_1(k)} \sum_{x=0}^k x p_x \quad (2.4)$$

$$m_2(k) = \frac{1}{P_2(k)} \sum_{x=k+1}^{L-1} x p_x \quad (2.5)$$

The cumulative mean upto level  $k$ :

$$m(k) = \sum_{x=0}^k x p_x \quad (2.6)$$

Let  $m_G$  is the mean of all the intensity values of image. Then the *between-class variance*  $\sigma_B^2$ , is given by-

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \quad (2.7)$$

Iterating through all the values for  $k$  such that for certain value of  $k$  the  $\sigma_B^2$  maximizes. That value will yield the optimum threshold point. This method assumes that the image is uniformly illuminated and histogram of the image is bi-modal. The method is suitable for segmenting images which have uniformly colored background.

### 2.2.2 k-Means Method of Image Segmentation

K-mean method is a user interactive method used for the segmentation of the image. This method segments the image into K clusters i.e. K segments. The number of segments has to be supplied by the user either directly or using some method. These are called seed points and form the cluster center. The basic algorithm of k-mean clustering is given by-

1. Mark the K seed points in the image using some technique.
2. Calculate the distance of each pixel with all the cluster center and assign that pixel to that cluster which has the minimum *distance* with that cluster center.
3. Update all the cluster centers by simply averaging all the values of the pixels in that cluster.

4. Iterate steps 2 and 3 unless the cluster centers stops changing.

### 2.2.3 Graph Cuts

Interactive segmentation is one of the most popular method of Image Segmentation. In this method of segmentation there is a minimal (though) necessary involvement of user, as there can be no better judgment by computer so as to satisfy human. The paper [13], describes the segmentation based on the graph cut technique. In this technique of segmentation, the user marks the certain set of pixels which are part of subject of interest as “foreground” and the “background” and based on the information supplied. These are called “hard constraints”. They serve as clue to machine in determining the segmentation. This segmentation is based on calculating the global optimum satisfying the hard constraints. The cost of segmentation is defined in terms of boundary and region properties of the segments. They serve as soft constraints for segmentation. The cost function used here is general enough to include regional and boundary properties.

Let  $P$  be the set of pixels in a neighborhood system  $N$ . Let  $(p, q)$  be the unordered pair of neighborhood system in  $N$ . For the  $P$  pixels, consider a vector  $A = (A_1, A_2, \dots, A_p, \dots, A_P)$  be a binary vector whose component  $A_p$  are the assignments to pixel  $p$  in neighborhood  $P$ . Each component of vector  $A$ , signifies either “foreground” or “background”. The vector  $A$  implies the segmented pixels. Then “soft constraints” that are imposed on boundary and region of  $A$  are described by the cost function  $E(A)$ :

$$E(A) = \lambda \cdot R(A) + B(A) \quad 2.8$$

where

$$R(A) = \sum_{p \in P} R_p(A_p) \quad 2.9$$

$$B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p, A_q) \quad 2.10$$

The  $R(A)$  term specifies the regional properties and  $B(A)$  specifies the boundary properties term. The term  $\lambda$  species the relative importance of region wrt. boundary. The regional term describes the penalties for assigning any pixel  $p$  as the object or background.

The term  $B(A)$  assumes the “boundary” properties of the segmentation. The coefficient  $B_{\{p,q\}}$  is always positive. This coefficient describes the penalty for the discontinuity between neighboring pixels  $p$  &  $q$ .  $B_{\{p,q\}}$  is large when pixel  $p$  and  $q$  are similar and is very small (nearly zero) when pixel are different. A suitable choice is cost based on the local intensity gradient.

The reasons for marking of those pixels is twofold, the pixels which provide hard constraints are assumed to be either part of foreground or background, and they provides intensity distribution of foreground or background pixels.

**Segmentation Technique:** Let a graph  $G = \langle E, V \rangle$ . Here  $E$  is the number of undirected edges in the graph and  $V$  be the number of vertices. Each edge  $e \in E$  in the graph is associated with a corresponding weight  $w_e$ .  $V$  is the set formed from the pixels in the image. Addition to these pixels (forming nodes of the graph) two additional nodes are also included labelled as *source* and *terminal*. A graph cut is a cut on graph that separates the source and edges. The cost of cut is given by

$$|C| = \sum_{e \in C} w_e \quad 2.11$$

Where  $w_e$  is the weights of the edges and  $C$  is the cost of cut which is the sum of weight of those edges that are involved in the cut.

Let a set denoted by  $O$  represents the subset of nodes marked as “object” and set  $B$  be the set denoting “background” nodes. These nodes are the *seeds* points. These seed points are marked by either users, or using some predefined criterion. These seed are used in the computation of as *hard constraints* to find the global minimum of the equation (2.7) to find the segmentation  $A$  such that

$$\forall p \in O, \quad A_p = "obj"$$

$$\forall p \in B, \quad A_p = "bkg"$$

The nodes  $V$  in graph  $G$  are formed from the image pixels  $p \in P$ . Additional to these nodes, two more nodes marked as  $S$  (for source connected pixels) and  $T$  (for sink connected pixels) are also added, such that the set of nodes is given by:

$$V = P \cup \{S, T\}$$

The edges  $E$  consists of two types of edges. For a pixel  $p \in P$  the edges which are connected to source  $\{p, S\}$  and sink nodes  $\{p, T\}$  are called  $t$ -links. And for pixels  $q \in P$ , edge between  $\{p, q\}$  such that  $\{p, q\}$  belong to neighborhood  $N$ , the links are called  $n$ -links. The table (2.1) gives the detail assignment of weights-

Edge connections	Cost of links	Conditions
$\{p, q\}$	$B_{\{p,q\}}$	$\{p, q\} \in N$
$\{p, S\}$	$\lambda \cdot R_p("bkg")$	$p \in P, p \notin O \cup B$
	$K$	$p \in O$
	$0$	$p \in B$
$\{p, T\}$	$\lambda \cdot R_p("obj")$	$p \in P, p \notin O \cup B$
	$0$	$p \in O$
	$K$	$p \in B$

Table 2.1, Edge Weights Assignment

The function  $R(.)$  gives the regional penalties. One of the function for the assignment of regional penalties is given by:

$$R_p("obj") = -\ln P_r(I_p|O)$$

$$R_p("bkg") = -\ln P_r(I_p|V)$$

This function is based on the intensities of the pixels marked by the seed points.  $R_p("obj")$  gives the conditional probability distribution of pixels when the pixels are marked by foreground. Similarly  $R_p("bkg")$  gives the conditional distribution of intensities marked by background.

The boundary penalty between two neighboring pixels is given by:

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p, q)}$$

For the pixels with similar intensities, there is assigned a very high value of weight and for the discontinuities, the assigned cost is low. However if the intensity difference is very large ( $I_p - I_q > \sigma$ ), then the assigned cost is small.

For the minimization of equation (1), the *maxflow - mincut* algorithm is used. This algorithm is introduced by Ford-Fulkerson and is used to minimize the flow in a network.

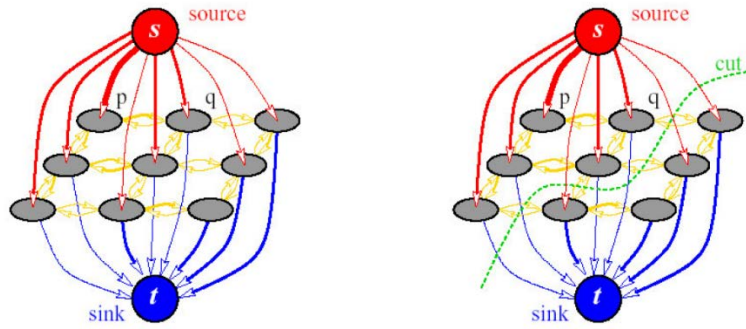


Figure 2.2, Showing graph cut for 3x3 image

Figure 2.2 shows an example of a graph. The image () shows the graph  $G$ . The  $T$  and  $S$  are the source and terminal nodes. The edge weights represents the regional terms (2.8) and the boundary terms (2.9) of the cost function. The assignments is given in figure (a). Note the pixels marked with  $B$  and  $O$  are seed points. Once the seeds are marked and weights are assigned, computation for the minimization of equation (2.7) is done. After finding the global minimum, the graph is cut. This cut is such that the cost of cut is

A cut in the graph is given in figure (2.2). The cut completely separated the source and terminal connected pixels. The minimum cut gives the minimum cost of segmentation  $C$ .

## **Chapter 3: Computer Vision**

### **Computer Vision**

### 3.1 Introduction

A human brain understands things much clearly by the seeing than by any other means and thus images fulfils the purpose very well. An image is a method of representation of any subject of interest, by presenting a visual depiction of subject in a two dimensional surface (paper or a display screen) or in three dimension (a statue). Presently a digital image is a two dimensional representation of images on the digital screen (say a computer).

Computer Vision: As the name suggests, computer vision is the entire process which by which a computer *understands* the image. By understanding it means, the computer analyzes the image, processes the image, extract the information and produces the results based on the information gathered from the image. It is a part of artificial intelligence and the *vision* is like seeing things by animals. The computer vision involves steps such as acquiring the image from an input device (cameras, scanners, sensors, etc.), analyzing based on algorithms and produce symbolic information.

In the digital image processing, the contents of the image are the source of information. To extract the information the foremost step is to the segment the image. The segmentation is the process of portioning the image into distinct contents based on the interest of user. The region of interest is called the foreground and the region which is not part of interest is called the background of image. This foreground part is separated from the background part and background part is discarded. Then the visual features of the foreground are analyzed. The information extracted from the analysis is then quantified. The quantification of information is done by *descriptors*. The descriptors is a terminology given to entire process of analyzing the content of foreground based on certain algorithm and produce symbolic information. This symbolic information can be used further as the application is required, for example this information can be used to identify one class of objects from different class of objects, etc.

The description techniques are based on the shape of the object of interest, the intensity (or color information) of the object, the pattern of texture on the object of interest, or any statistical data. There are a number of object description techniques already available in literature.



## 3.2 Shape Descriptors

The shape descriptor is a type of descriptors which describes the geometrical shape of the object of interest. This helps in statistical analysis of shapes and the comparison of one shape with the other. A large number of shape descriptors are already available in literature. Depending upon the problem the shape descriptor must be selected. For a proper shape descriptor, it must satisfy the property of shape invariance, i.e. shape description must be independent of rotation of object. It must be clearly able to distinguish one shape from other, but should not discriminate between shapes of same class. It must be computationally efficient and produce results quickly and effectively. A large number of shape descriptors are already available in literature such as MPP, Fourier Descriptors, HOGs (Histogram of Oriented Gradients), etc. One such descriptor is shape context.

### 3.2.1 Shape Context

*Shape Context* is a shape descriptor proposed by S. Belongie and J. Malik in his paper “*Matching with Shape Contexts*” [7]. This shape descriptor is simple and yet has proved to be a very powerful descriptor for describing the shape. In this paper, he presented a complete framework for matching shapes. This shape context technique is rotation invariant and is proved to be an effective technique compared to other shape descriptors. The shape context is based on measuring 2 dimensional histogram for each sample of the shape object so as to measure the relative position of each pixel with respect to other. The procedure for the identification of shape by shape context is discussed thoroughly in following context.

#### *Shape Context Steps:*

1. Extract the boundary of object and randomly select ‘n’ points from the entire sample space ‘N’. The number of points should be so selected that they should represent the entire shape. If the value ‘n’ is equal to ‘N’, i.e. all samples are selected then, the representation of shape becomes exact shape. Thus the set  $P = \{p_1, p_2 \dots p_n\}$  is the set of edge pixels of the object. The  $p_i \in R^2$  must be satisfied i.e.  $p_i = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix}$ . Where  $x_i$  and  $y_i$  are location of pixels. The aim of the shape matching is to find best match for points  $p_i$  in first shape with points  $q_i$  in the second shape.

2. For the first set of points  $P$ , the relative position of each  $p_i^{\text{th}}$  pixel is measured with respect to remaining  $n - 1$  pixels. To calculate this, the log-polar histogram is calculated. This histogram represents the relative position of the  $i^{\text{th}}$  pixel with remaining pixels.

$$h_i(k) = \text{hist}(P, r, \theta) \quad (3.1)$$

where,  $h_i(k)$  is the histogram of  $k$  bins in 2-D,  $r$  is the logarithmic distance step from point  $p_i$  and  $\theta$  is the angular steps. This is represented in figure 3.1.

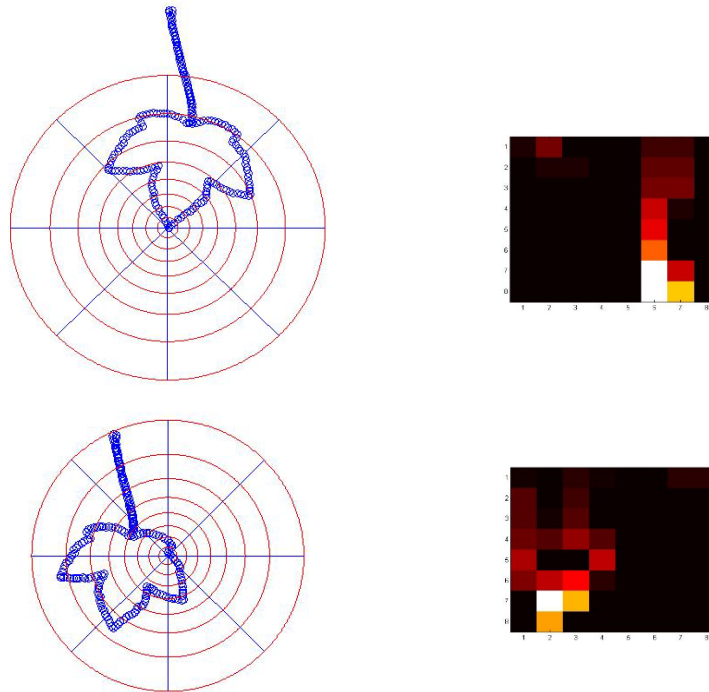


Figure (3.1) a, b, c & d. (clockwise). For a pixel  $p_i$  (shown as tip of leaf shape in figure a) the 2D log polar histogram is shown at right. Similarly for another pixel figure d, the histogram shown at right (figure c). As it can be seen visually, the histograms for two positions are clearly distinct.

3. Similar steps are followed for the second shape. For the second shape a set of points  $Q = \{q_1, q_2, \dots, q_m\}$  is sampled. A similar histogram  $g_i(k)$  for the second shape is calculated for every sample in second shape. The aim of shape context is to find for every  $p_i$  in first shape, a unique  $q_j$  in second shape such that there is a one to one matching of points from one shape to another.

4. The cost for histogram matching is explained as following: assume  $C_{ij}(p_i, q_j)$  be the cost of matching of a sample  $p_i$  with  $q_j$ . Then this cost is calculated by comparing histograms by *Chi-Squared Test*. The cost of matching is given by:-

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - g_j(k)]^2}{h_i(k) + g_j(k)} \quad (3.2)$$

Where,  $h_i(k)$  and  $g_j(k)$  are the 'k' bin histograms of  $p_i$  and  $q_j$  respectively.

5. Now, the cost of matching of histograms  $C_{ij}$  for all the pair of samples  $p_i$  from first shape and  $q_j$  from the second shape is available. The need is find each unique pair of  $p_i$  and  $q_j$  such that the overall cost of matching of minimized provided matching is one-one. The problem is similar to that of '*travelling salesman problem*' and Hungarian method of optimization can be used to solve such problem.
6. Once the problem of one on one correspondence is solved, we need to align the shapes. This is a done to achieve shape invariance matching. Given a set of corresponding points between two shapes, the need is to estimate transformation parameters so as to match any arbitrary points from one shape to the other. This problem is called *Image Registration* problem in literature.

Given a set of control points, image registration is used to find corresponding scenes between to images. This type of method is used in images where image of same object of interest is taken from different angles or distances, or using different illuminations, etc.

Given a set of ordered pairs of point

$$\{(p_i, q_j(i)), \quad i \in 1, 2, \dots, N\}$$

We want to find transformation  $f(p_j)$  that satisfies:

$$q_j(i) = f(p_j)$$

The similarity transformation is a transform that is used for mapping the global translation, rotation, scaling and shearing effect of the images. Consider,

$$T(x) = Ax + o$$

then affine transformation parameters be obtained as:

$$T = (A, o)$$

Where,

$$o = \frac{1}{n} \sum_{i=1}^n (p_i - q_j(i)) \quad (3.3)$$

$$A = (Q^{-1}P)$$

Where  $Q^{-1}$  is the pseudo inverse of Q.

And P is the ordered pair of samples:

$$P = \begin{bmatrix} 1 & p_{11} & p_{12} \\ \vdots & \vdots & \vdots \\ 1 & p_{n1} & p_{n2} \end{bmatrix}$$

And similarly for Q.

7. Once the transformation of points of set Q is done, the distance measure between the two shapes can be given as:

$$D(P, Q) = \frac{1}{n} \sum_{p \in P} \arg \min_{p \in P} C(p, T(q)) + \frac{1}{m} \sum_{q \in Q} \arg \min_{q \in Q} C(p, T(q)) \quad (3.4)$$

Where T(.) denotes the transformed points of q by transformation T().

### 3.3 Texture Descriptors

Texture descriptors are those descriptors which describe the texture pattern on the object. Though there is no formal definition of texture, in broad way texture can be described as: The spatial arrangement of intensities in a part of image in a repetitive manner so as to appear as a work of art. From the statement, it can be deduced that in digital images, the repetition of certain pattern of values in in the matrix. These textures prove to be very helpful in discriminating one object from other and hence are used in the image segmentation process.

There are two ways to deal with the textures in images: first is by quantitative measurement of intensity pattern by statistical approaches, another method is by comparing the textures in the image with the small units of textures (called *texels*). Presently in literature there a number of

methods of describing textures are available. Texture analysis by invariable moments is a statistical approach which is very helpful in finding contents which are not very closely related. One of the famous approach of structured texture recognition is texture recognition by *Gabor filters*. Gabor filters have been extensively used in texture analysis as the closely model the human vision system. Then there is another method of texture recognition is by analyzing texture by Local Binary Patterns (LBP). The LBP operator was first introduced by *Ojala et. al* in the paper for texture classification [8]. In the paper it was shown that the statistical approach and other methods of texture recognition were computationally expensive as compared to his proposed method. A large number of papers related to LBP have already been published. Applications such as face recognition [4], image retrieval [3], object tracking, biomedical image processing, etc., have already shown the effectiveness of the descriptor. The method of texture identification by Local Binary Pattern (LBP) is discussed as further:

### 3.3.1 Local Binary Patterns

The LBP operator was introduced by Ojala et al. in his paper: [8]. Consider a 3x3 image as shown the figure 3.2. In the figure the pixel at (2, 2) whose intensity value is given as 64. This is the center pixel. This center pixel is compared with all the remaining pixels i.e. with (1, 1), (1, 2) ... (3, 3). If the weight of the center pixel is more than the neighboring pixel then the weight allotted to that pixel is 0, else 1. This is shown in figure 3.2 (b)

23 <sub>(1,1)</sub>	105 <sub>(1,2)</sub>	64 <sub>(1,3)</sub>	0	1	1
97 <sub>(2,1)</sub>	64 <sub>(2,2)</sub>	32 <sub>(2,3)</sub>	1	CP	0
43 <sub>(3,1)</sub>	30 <sub>(3,2)</sub>	255 <sub>(3,3)</sub>	0	0	1

Figure 3.2 (a) and (b) 3x3 grid and grid after comparison with CP

After this the value for the center pixel is calculated by equation (3.6):

$$LBP_{(P,R)} = \sum_{i=1}^P 2^{(i-1)} \times f(g_{i|R} - g_C) \quad (3.6)$$

Where,

$$f(x) = \begin{cases} 1, & x \geq 1 \\ 0, & \text{else} \end{cases} \quad (3.7)$$

In the above expression, P is the number of points uniformly sampled around a circle of radius R. For 3x3 matrix, P value is 8 and R value is 1. In general the P, R values can be selected arbitrarily. This can be seen in the figure 3.3.

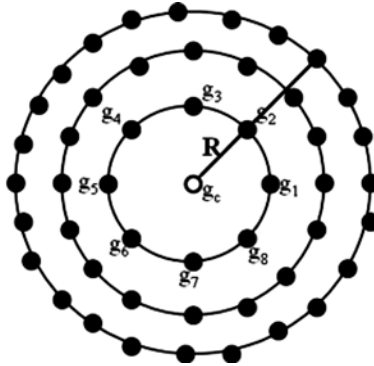


Figure 3.3 showing different P,R combinations

	4	2
16		
		128

	150	

Figure 3.4

For the 3x3 matrix shown in figure 3.4, after the weighting process, the LBP value for CP is given as 150. This basic LBP is not rotation invariant. The rotation invariance for the Local Binary Pattern was shown in the paper [8]. This paper presents the *Uniform binary Patterns* which is the extension of LBP. An LBP pattern is called uniform pattern “U”, if the number of bitwise transitions from 0 to 1 or vice versa are not more than 2. For example, consider a pattern given as 00000000 (taken around a circle, starting from anywhere around circle), here there is no transition. Consider another pattern 00011100. In this pattern the number of transitions from one to zero and from 0 to one are 2. While the pattern 10110110 has six transitions, and hence is considered non uniform. Among all the uniform binary patterns, each uniform binary pattern is assigned a separate

label, while all the non-uniform patterns are grouped under one label. The number of different labels for  $P$  sampling points is given as:  $P(P-1) + 3$ . For example, for  $3 \times 3$  matrix, ( $P = 8$ ,  $R = 1$ ), the number of distinct labels is 59. There are two main reasons of using only uniform patterns. First is that the robustness of LBP increases as uniform patterns are less susceptible to noise. Secondly, the uniform patterns are used to achieve rotation invariance.

### **3.4 Application to Project**

The project utilizes the LBP as texture descriptor and Shape Context as a shape descriptor. Combined the methods of segmentation and above descriptors forms the crux of the system. Prior to the feature extraction and after segmentation, the image of the object is aligned and scale. This has been described in details in chapter 5. On comparing the results with the paper [6], which utilizes the GLCM and Gabor filters for texture identification, and curvelet transform for shape identification it can be seen that the above operators performs better and provide more accuracy.

## **Chapter 4: Pattern Recognition**

### **Pattern Recognition**



## 4.1 Introduction

Machine learning is a branch of engineering that helps the machine to recognize the patterns. This pattern recognition problems find applications in the field of speech recognition, fingerprint recognition, optical character recognition, DNA sequence recognition, etc. For the image identification process, it is best to design a system that works based in the same way as nature recognizes the patterns.

Any recognition system is based on the mathematical *model* and the *features* selected. Feature is a distinctive attribute of the sample which helps in recognition process. For example, in the plant recognition system, the feature of leaf images can be leaf shape, leaf texture, venation pattern, etc. These features are mathematically coded and are fed to the *models* for the identification.

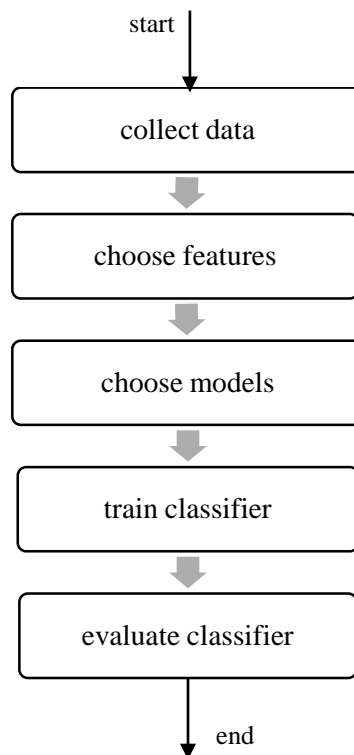


Figure 4.1, The design of pattern recognition system. [14]

The entire design cycle can be understood by figure 4.1. The first step is to gather samples. The data collection is done using some transducer that converts the real world data (e.g. microphone camera) into digital data (recordings, images, etc.). The performance of the system depends on the samples used for the training of the system hence the data collected must be in some sense uniform

and only *distinctive features* must vary (e.g. images taken of leaf samples must be uniformly lit, camera must introduce minimum noise, the resolution of images must be high, the background must be clear).

The feature selected for identification must be invariant to deformations. In the plant recognition system, it was required that the feature must be scale invariant, rotation invariant, translation invariant, projection invariant. Hence such *feature extractors* were used which satisfies the above properties.

The choice of the features that are used for the pattern recognition system is important. As stated, only distinctive features must be used. An obvious choice is to select those features that are easy to extract, insensitive to noise, follow above requirements of invariance and distinctive for each class of samples and similar for one class of samples.

The model used for classification plays an important role. Their proper selection over samples of data can produce vary different results. One such choice of classification is using neural networks. The neural networks attempts to solve the problem as they are solved in natural by humans. This method is used in the plant identification system.

## 4.2 Neural Networks

Neural network are largely based on how neurons in brain work. They are used in the problems of identification of image Classification, neural language processing, and CAT classification. Difference between problem of Regression and Classification is classification of Regression is used for analog classification while Classification is used for discrete classes of data.

For example, let there is a problem is to estimate on test scores based on the hours studied and hours slept the night before the exams. Let the data pair be given as (X,Y): (3,5; 5,1; 10,2) and the marks scored be (75,82,93) then how many marks will be scored if one studied for 8 hours and slept for 3 hours. The first step is to scale the data, as neural networks on the normalized data.  $X_{\text{norm}} = X/\max(X)$ ;  $Y_{\text{norm}} = Y/100$ .

A neural network has basically has a single input layer, output layer and intermediate layers. Intermediate layers are called as **hidden layers**. When there are many and many hidden layers, these are known as deep networks giving rise to the term: **deep learning**.

In neural network, circle represents neurons and lines are analogues to synapses. In machine learning, synapses mean nothing but multiplying weights.

Neurons job is to act together the output from other synapses and apply an activation function. Certain activation function allow neural network to model complex non-linear patterns, that simpler models may miss. Some famous function are sigmoidal function, hyperbolic tangent function, rectified linear function.

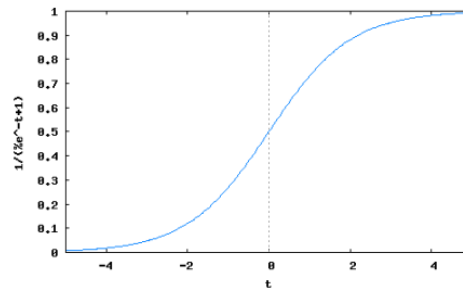
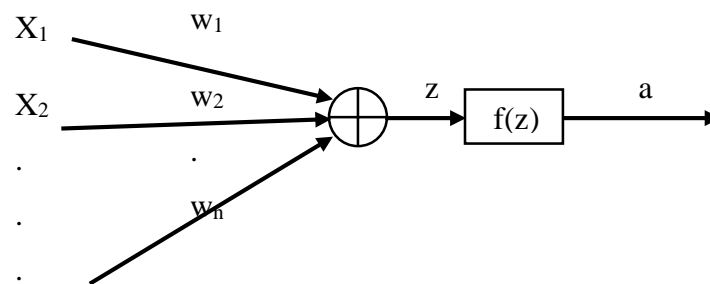


Figure 4.2 Sigmoidal Activation Function

Sigmoidal function is given by:  $f(a) = 1/(\exp(-a))$ , plot is shown in figure 4.1. Properties of sigmoidal function: squashes the neuron's pre activation between 0 and 1, always positive, bounded, strictly increasing function.



$$z = X_1 + X_2 + X_3 + \dots + b_n$$

Figure 4.3, A simple neuron

For the above problem, the neural network can be shown as:

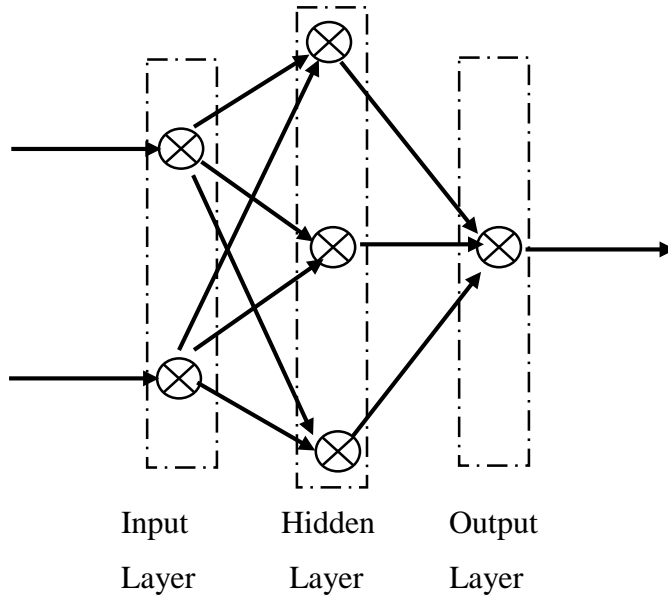


Figure 4.4, 2-Layered Neural Network with 2 nodes in input layer, 3 in hidden layer and 1 in output layer

For the relation between input layer and hidden layer for the above example, the matrix representation can be stated as:

$$\begin{bmatrix} 3 & 5 \\ 5 & 1 \\ 10 & 2 \end{bmatrix} \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} = \begin{bmatrix} z_{11}^{(1)} & z_{12}^{(1)} & z_{13}^{(1)} \\ z_{21}^{(1)} & z_{22}^{(1)} & z_{23}^{(1)} \\ z_{31}^{(1)} & z_{32}^{(1)} & z_{33}^{(1)} \end{bmatrix}$$

or in summary the output for second layer can be calculated as:

$$z^{(2)} = X \cdot W^{(1)} \quad (4.1)$$

Where (2) denotes the activity for the second layer.

Z is of size 3x3

To  $z_{3 \times 3}^{(2)}$ , we apply the activation function. This can be applied independently to each entry in the matrix  $z^{(2)}$ .

For the forward propagation, the activity for the second layer can be modelled as shown in equation (4.2) and  $a^{(2)}$  is of same dimension that of  $z^{(2)}$  i.e. 3x3.

$$a^{(2)} = f(z^{(2)}) \quad (4.2)$$

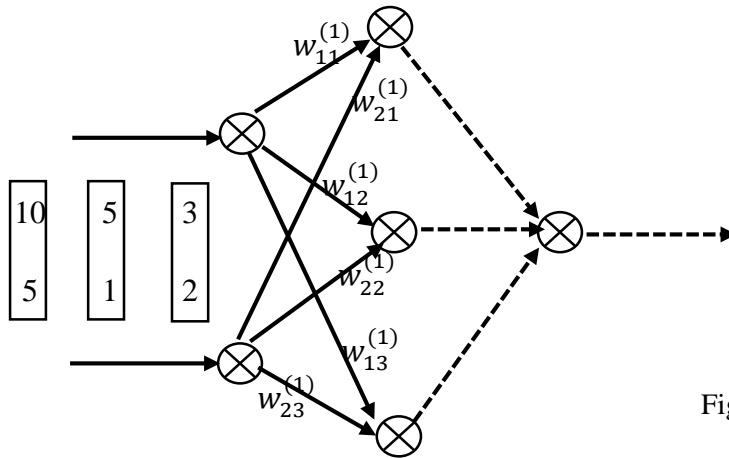


Figure 4.5, Figure showing first layer activity

where 'f(.)' is the applied activation function.

Now, to finish the propagation,  $a^{(2)}$  has to propagate all the way up to  $\hat{y}$ .  $\hat{y}$  is the initial estimate of the given problem. This is shown in equation (4.3)

$$z^{(3)} = a^{(2)}w^{(2)} \quad (4.3)$$

Where  $z^{(3)}$  is the activity for the third layer. To equation (4.3) one more activation function has to be applied.

$$\hat{y} = a^{(3)} = f(z^{(3)}) \quad (4.4)$$

Now for the above example  $w^{(2)}$  is a 3x1 matrix given as  $[w_{11}^{(2)} \quad w_{12}^{(2)} \quad w_{13}^{(2)}]$

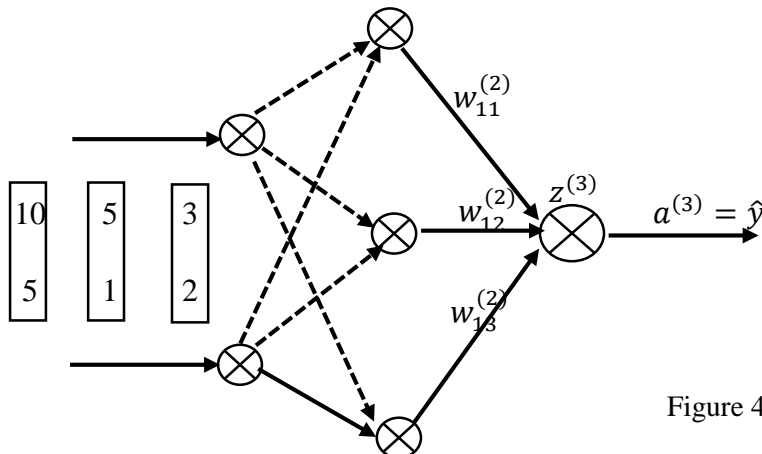


Figure 4.6, Figure showing second layer activity

### Training the Neural Network

Cost function: cost function allows us to express exactly how wrong the model is. The training of the neural network actually refers to the process of minimization of the cost function. The cost

function is the function of two parameters: our input data (for the above example hours slept and hours studied) and weight on each synapse. The target of training the neural network is to minimize the costs of changing weights.

The cost function is given as:

$$J = \sum \frac{1}{2} (y - \hat{y})^2 \quad (4.5)$$

from the equations (4.1), (4.2), (4.3) and (4.4), equation (4.5) can be modified as:

$$J = \sum \frac{1}{2} (y - f(f(X \cdot W^{(1)}) \cdot W^{(2)}))^2 \quad (4.6)$$

For finding the error the derivative  $\frac{\partial J}{\partial W}$  is calculated. The target is to minimize the  $\frac{\partial J}{\partial W}$ .

For the above problem  $\frac{\partial J}{\partial W}$  is split into two parts:  $\frac{\partial J}{\partial W^{(1)}}$  and  $\frac{\partial J}{\partial W^{(2)}}$  as there are two layers in the network.

$$\frac{\partial J}{\partial W^{(1)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(1)}} & \frac{\partial J}{\partial W_{12}^{(1)}} & \frac{\partial J}{\partial W_{13}^{(1)}} \\ \frac{\partial J}{\partial W_{22}^{(1)}} & \frac{\partial J}{\partial W_{23}^{(1)}} & \frac{\partial J}{\partial W_{24}^{(1)}} \end{bmatrix} \quad (4.7)$$

&

$$\frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} \frac{\partial J}{\partial W_{11}^{(2)}} \\ \frac{\partial J}{\partial W_{21}^{(2)}} \\ \frac{\partial J}{\partial W_{31}^{(2)}} \end{bmatrix} \quad (4.8)$$

Equation (4.7) can be solved as:

$$\frac{\partial J}{\partial W^{(2)}} = \sum \frac{\partial \frac{1}{2} (y - \hat{y})^2}{\partial W^{(2)}} \quad (4.9)$$

for the sake of simplicity, it is assumed that  $W$  is a single element, then equation (4.9) becomes:

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial W^{(2)}} \quad (4.10)$$

Differentiating equation (4.4), by chain rule differentiation, equation becomes-

$$\frac{\partial \hat{y}}{\partial W^{(2)}} = \frac{\partial \hat{y}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial W^{(2)}} \quad (4.12)$$

Using equation (4.12) in equation (4.11)-

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial z^{(3)}} \cdot \frac{\partial z^{(3)}}{\partial W^{(2)}} \quad (4.13)$$

In the above equation,  $\hat{y}$  is simply  $\hat{y} = f(z^{(3)})$ . If  $f(z)$  be the sigmoid activation function as:

$$f(z) = \frac{1}{1 + \exp(-z)}$$

then its derivative can be given as-

$$f'(z) = \frac{\exp(-z)}{1 + \exp(-z)} \quad (4.14)$$

Using equation (4.14)  $\frac{\partial J}{\partial W^{(2)}}$  can now be written as:

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y}_0) \cdot f'(z^{(3)}) \cdot \frac{\partial z^{(3)}}{\partial W^{(2)}}$$

and from equation (4.3), it can be seen that there is a linear relationship between activation  $a^{(2)}$  and weight on synapse  $w^{(2)}$ . Hence  $\frac{\partial z^{(3)}}{\partial W^{(2)}}$  is simply the activation weights 'a' on the synapse.

### ***Backpropagation***

Another view about neural network is that, the error is backpropagation at the each weight by multiplying with the activity of each weight. The weight that contributes more to error will have larger activation, hence larger  $dJ/dW$  values and will be changed more in the gradient descent. This way of training the neural network in where weights are changed in accordance with the backward propagating error is called backpropagation training.

For all the weights for second layer

$$\frac{\partial J}{\partial W^{(2)}} = - \sum (y - \hat{y}) f'(z^{(3)}) \frac{\partial z^{(3)}}{\partial W^{(2)}} \quad (4.15)$$

Let,

$$\delta^{(3)} = - \sum (y - y_0) f(z^{(3)})$$

be defined as backpropagation error, then for above example, simplified matrix representation for equation (4.15) is-

$$\frac{\partial J}{\partial W^{(2)}} = \begin{bmatrix} a_{11}^{(2)} & a_{13}^{(2)} & a_{13}^{(2)} \\ a_{21}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} \\ a_{31}^{(2)} & a_{32}^{(2)} & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} \delta_1^{(3)} \\ \delta_2^{(3)} \\ \delta_3^{(3)} \end{bmatrix}$$

Similarly for the calculation of  $\frac{\partial J}{\partial W^{(1)}}$ , from the equation (4.8)-

$$\begin{aligned} \frac{\partial J}{\partial W^{(1)}} &= \sum \frac{\partial \frac{1}{2} (y - y_0)^2}{\partial W^{(1)}} \\ &= - \sum (y - y_0) f'(z^{(3)}) \cdot \frac{\partial z^{(3)}}{\partial W^{(1)}} \\ &= - \sum \delta^{(3)} \frac{\partial z^{(3)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial W^{(1)}} \end{aligned}$$

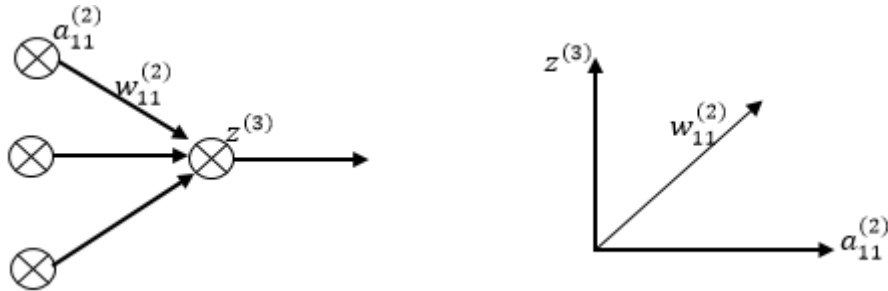


Figure 4.7, Figure showing linear relation between third layer output and output of activation function for a node in intermediate layer

Activation weights and synapse weights are linearly related, hence  $\frac{\partial J}{\partial W^{(1)}}$



$$\frac{\partial J}{\partial W^{(1)}} = \delta^{(3)} \cdot (W^{(2)})^T \cdot f'(z^{(2)}) \cdot \frac{\partial z^{(2)}}{\partial W^{(1)}} \quad (4.16)$$

The term  $\delta^{(3)}$  is the backpropagating error,  $(W^{(2)})^T$  is simply the weights of the intermediate layer. Term  $\frac{\partial z^{(2)}}{\partial W^{(1)}}$  is again linearly related. Hence all the terms are known. In matrix notation, equation (4.16) can concisely be written as:

$$\frac{\partial J}{\partial W^{(1)}} = X^T \delta^{(3)}$$

where  $X^T$  is the input to neural network.

### 4.3 Application to Project

A similar MLP neural network with backpropagation training is used in the training of the system is used. The choice is made because of the underlying fact that the real world images must be perceived by the system must be in the same way as human brains treat the images. For the practical problems, two layer model is sufficient for most of the methods. Determining the number of perceptron in the intermediate is a method of trial and error. Details of the system used in the implementation of system is discussed thoroughly in the chapter 5.

## **Chapter 5: Plant Identification Framework**

### **Plant Identification Framework**

## 5.1 Proposed Model

As describe in chapter one, the project is about the identification of the plants on behalf of the image of the leaf. All the technique from image segmentation to classification as described in the chapter one to four are used in the project. The project maybe assumed as similar to that of projects of face identification or finger print recognition, but plant identification problem is more complicated than these. It is so because the overall shape of and the position of visual features remains same. For example, in face recognition, the position of eyes will remain fix relative to that of nose. But in plant recognition, a single plant may have variety of leaf differentiated shapes or color. Hence the problem becomes slightly difficult.

The proposed method framework to solve the problem is shown in figure: 5.1

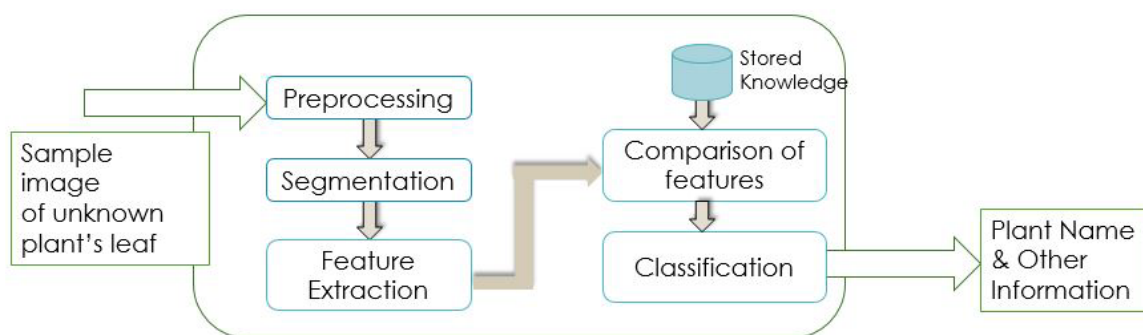


Figure 5.1, Proposed Model Block Diagram

The sample input is taken from the camera and is fed to the system. The steps of preprocessing are applied and the leaf part (which is object of interest) is segmented from the background. Then the features of are extracted from the sample image. These features are compared with the stored knowledge and a decision is made by the system that which type of leaf it is. Or if the leaf is in database itself or not.

The project is divided into two phases of implementation: the training phase and the testing phase. In the training phase of the system, the process of database creation and storing the knowledge is done. Also the classification part of problem is done. And in testing phase, the system is tested. The details of the implementation is explained in step by steps in the following context:

## 5.2 Experiment

The sample input is taken from the camera and is fed to the system. The steps of preprocessing are applied and the leaf part (which is object of interest) is segmented from the background. Then the features of are extracted from the sample image. These features are compared with the stored knowledge and a decision is made by the system that which type of leaf it is. In the experiment, a database of 10 class of leaf images is used. Sample from each class of image is shown in figure-

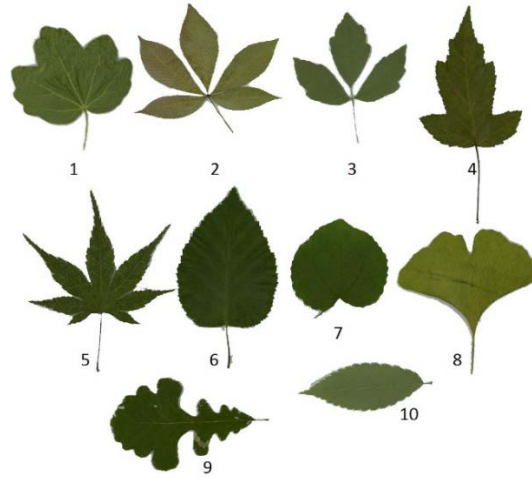


Figure 6: Leaf Samples from each class

Each class of image, has 20-30 images. For the image belonging to  $k^{\text{th}}$ , a set consisting of leaf boundary samples is stored as  $Q_k$ . The *shape matching cost* for matching the shape of query image with the  $k^{\text{th}}$  sample set is given as  $D_k$ .  $FV_1$  is the value of first component of feature vector  $F$  and is given as:

$FV_1 = \text{class index } (k) \text{ for which } D_k \text{ is minimum. Here } FV_1 \text{ is of unit length.}$

For the texture matching, the image is scaled and zero padded so that each image has dimensions of 999x999 pixels without deforming the original shape of leaf. Each image was then divided into 9 block (each is of size 333x333). To each block of image uniform LBP operator was applied with pixel radius and neighborhood combination  $(R, P) \in \{(1,8), (2,12), (3,16)\}$ . Hence for each block of image, histograms of length 59, 135 and 243 are obtained (since no. of bins of histogram =  $P(P - 1) + 3$ ). All the three histograms are then concatenated for each block of image. Such

histograms for all nine blocks are concatenated to form a feature length of 3933. This 3933 bins histogram is used as a feature vector  $FV_2 = \text{histogram of 3933 bins}$ .  $FV_2$  is of 3933 length.

The third feature vector is the original length of the leaf  $FV_3 = \text{leaf length}$ . And last feature vector being  $FV_4 = \text{leaf width}$ .

The overall feature F vector is obtained by concatenating each feature-

$$F = \{FV_1, FV_2, FV_3, FV_4\}$$

## 5.3 Results

### 5.3.1 Image Preprocessing and Segmentation

The implementation of the project starts with pre-processing of the leaf samples. The images of the sample are aligned in one direction. The alignment is based on the major axis of the leaf. A user interactive method of alignment is used. The original dimensions of the object of interest is also noted. The user roughly marks the tip of the leaf and base of the leaf. The horizontal axis is defined as the length from tip of the leaf to the base of the leaf. This horizontal axis is used for the alignment of the leaf. Once the leaf is aligned (with tip of leaf in upright direction), the image is segmented and the leaf area is extracted out form background. After segmentation, a bounding box is created around the leaf object. This gives the horizontal and vertical extent of the leaf (i.e. the length and the width of the leaf).

*Note*, for the measurement of proper dimensions, of leaf two precautions are required to take, first- the leaf stalk must be of standard size. Secondly- while gathering the sample, the images must be taken form a fixed distance for all sample of leaf. This is in practice a difficult task, hence some method of relative measurement must be used (e.g. while capturing the image of sample, the image of some object with known dimensions such as a coin, can be used for comparison).

In the training phase, the training of the system is done using the lab samples. These samples had clear background and in was in contrast with the leaf color. This helps in clear segmentation. Simple Otsu's method of segmentation can be used to separate out the foreground from background. The Otsu's method of segmentation is a global method of segmentation based on histogram thresholding. Thus the method of segmentation may consist of unconnected regions. As

on fact, the images of leaf are connected and a healthy sample may have nearly no holes within the shape of the image, a morphological operation of *opening* can be used to connect any unconnected regions, keeping the structuring element small. The requirement of keeping a small structuring element is so because, large structuring element may cause the loss of information of shape blades which is a vital information in identifying the leaf.

After this to separate the noise which is unconnected to the central leaf portion, a largest blob (part which has the maximum area) can be found. This resulting area presents the geometrical shape of the object. The entire process is shown in the figure 5.2.

It is to be noted that, in the testing phase, images are taken from the field samples. These images may not have a clear distinct background. Hence the Otsu's method of image segmentation is not suitable for segmentation purpose. The more robust method of graph-cuts is used for segmentation. This method is a user interactive method and produced far better results of segmentation than Otsu's method of segmentation. The method of *graph-cuts* is given in chapter-2 Figure 5.2 (b).

After the segmentation, the task is to extract the boundaries of the shape, from the binary segmented image. The four connected boundaries are traced by the method given in book *Digital Image Processing* [15]. These boundaries serves as input to the shape descriptor. The scatter plot of the sample leaf is shown in the figure 5.2 (c).



Figure 5.2 (a), (b) and (c); Segmentation of Image (a) is image in (b), and uniformly selected samples shown in figure (c)

### 5.3.2 Shape Matching

Once upon the shape is extracted, the next step is to extract the shape features. As described in chapter-2, the shape context descriptor is used to describe the shape of the object. Upon the extraction of boundary samples, the boundary is uniformly sampled so as to reduce the number of samples representing in the shape. Since using all the points would result in increase in the runtime exponentially.

For the boundaries shown in figure 5.2 (c), a log-polar histogram is calculated for every boundary sample. Figure 5.3 shows the calculated log polar histogram for the randomly selected sample. Similarly, another shape is processed and its boundaries are extracted in the same manner. Shape matching is done as the corresponding matching points have same log polar histograms, while unmatched shapes will have different histograms matching. This can be observed in figure 5.4

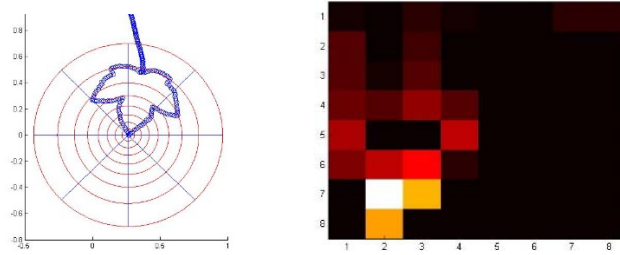


Figure 5.3, Log Polar histogram of a sample shown in left

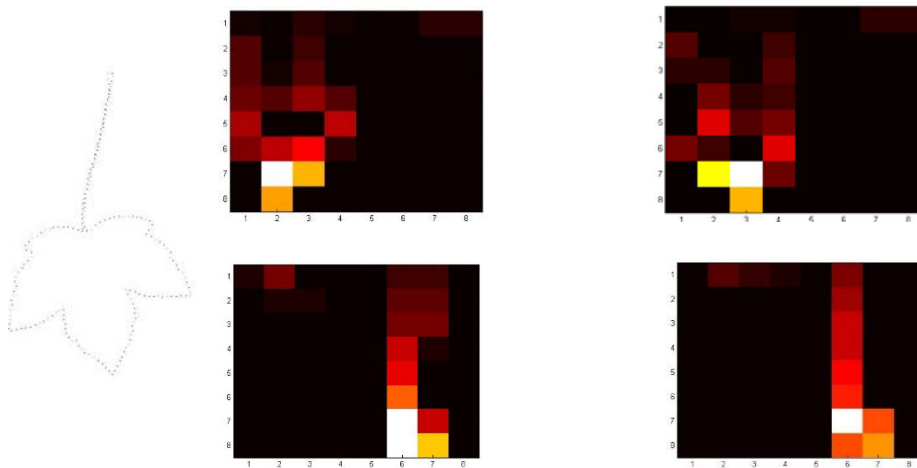


Figure 5.4,

After that histograms of all the samples for each shape is calculated, the task is to find corresponding matching points. This is done by comparing histograms for each sample in first

shape with all the histograms in second shape. The histogram matching is done by using chi squared test. The correspondence matching points between two shapes is presented in figure-5.5. After finding the matching points the shapes are aligned using a suitable transform. This is shown in figure.

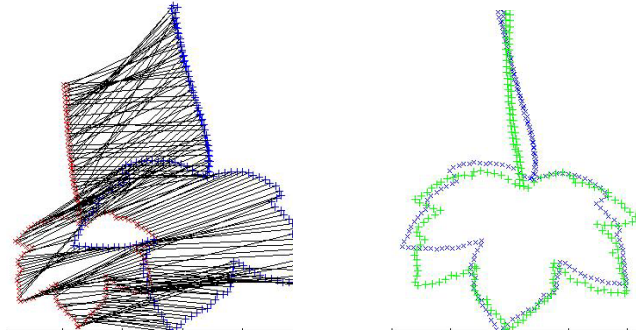


Figure 5.5 (a) One on one matching points, (b) Image of two objects after alignment

The final cost of shape matching after alignment gives the shape matching measure. It is this measure that decides that whether there is a match in shape or not. For the 10 classes of image, the matching cost of query image is calculated as  $D = \{D_1, D_2, \dots, D_{10}\}$ . If  $D_k$  is the minimum cost of matching, then the feature vector is assigned as  $FV_1 = k$ .

### 5.3.3 Texture Matching

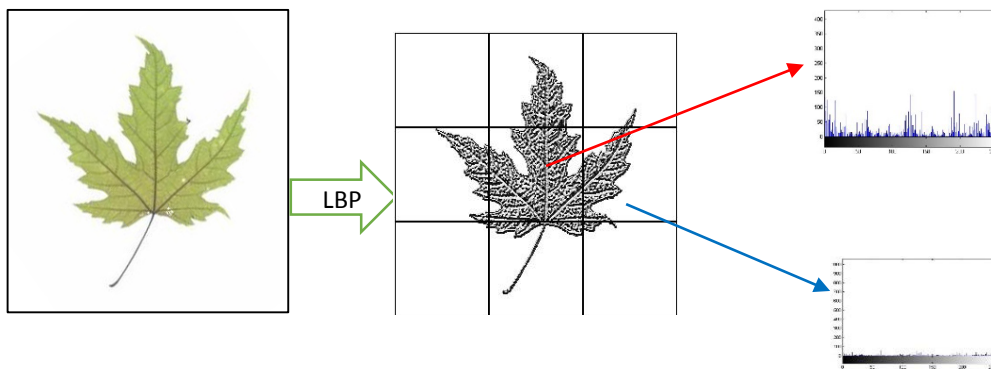


Figure 5.6, LBP image of given leaf. Different blocks have different histograms as can be seen from figure, in this way localization of features is achieved.

Another feature that is used in the identification process is the texture feature. The rotation invariant local binary patterns are used to identify the textures in the image. The local binary patterns brings out the information of venation patterns in the leaf. For the robust operation of LBP



operator, the entire image was divided into blocks of uniform sizes. Then the LBP operator was applied on each block. For each block, histogram is calculated.

This histogram represents the leaf. Any image with histogram, similar to the histogram of above image is considered as a match with image.

A similar database is created for each leaf. For each leaf, the LBP histogram is calculated and stored. The LBP histogram of query image is compared with the histograms of stored images, and a similar feature vector is formed with each element representing the cost of matching of query image with each shape stored in database.

For the training of the system, leaf samples for 10 classes were taken (one from each class). For each Image representing  $c^{th}$  class, Image is converted into grayscale image, background was removed, edge enhancement of image was done by unsharp filtering. The processed image was then scaled into a size of 999 x 999 pixels. Each image is then broken into 9 blocks, each block being of 333 x 333. For each block three LBP image was computed for the three type of neighborhood  $(R,P) \in \{(1,8), (1,16), (2,8)\}$ , where P being the number of samples uniformly taken around circle of radius R. If  $h_{blk,i}$  be the histogram of block  $i^{th}$  block, then block histogram is given by:

$$h_{blk,i} = \{h_{(1,8)}, h_{(1,16)}, h_{(2,8)}\}$$

where  $h_{(R,P)}$  is the histogram corresponding to LBP image of  $(R,P)$  neighborhood.

Histogram for the entire image is given by:

$$h_{img,c} = \{h_{blk,1}, h_{blk,1}, \dots h_{blk,9}\}$$

This histogram was stored and is used as the feature vector for the comparison purpose. This texture extraction was done for all 10 class of images.

The feature vector  $FV_2 = h_{img,c}$  and is of size 3933 bins.

## 5.4 Classification Results

The classification of leaf is done by Artificial Neural Network. A two layer perceptron model was taken for classification. Hidden layer had 10 nodes and output layer with 10 nodes. The above data

was classified using neural network toolbox in MATLAB® 2014b. Input to the neural network is the feature vector F. A total of 70% of the images are used for training the data, other 15% of the training images are used for validating the network for the generalization and to avoid the overfitting of the data. The rest 15% of the images are used for testing the network. These images were randomly chosen for training and testing from the entire database of images. The training of the *neural network* is done by extracting feature vector F from each training image. Sigmoid function were used as activation function in the intermediate nodes, while softmax activation function is used at the output nodes.

<b>Class Number</b>	<b>Species Name</b>	<b>Number of Samples Taken</b>	<b>Number of Samples Classified Correctly</b>
1	Acer campestre	27	27
2	Aesculus Galbra	30	28
3	Acer Griseum	20	18
4	Acer Ginnala	30	30
5	Acer Palmatum	30	28
6	Betula Jacquemontii	30	30
7	Cercidiphyllum Japonicum	30	30
8	Ginkgo Biloba	30	29
9	Quercus Macrocarpa	30	30
10	Zelkova Serrata	33	31

Table 5.1

The above data was classified using neural network toolbox in MATLAB 2014b, a total of 70% of the images are used for training the data, other 15% of the training images are used for validating the network for the generalization and to avoid the overfitting of the data. The rest 15% of the images are used for testing the network. These images were randomly chosen for training and testing from the entire database of images. The table 5.1 shows the classification results for the ten classes of images selected from the dataset provided by the website: *leafsnap* [16]. A total of 200

samples of images were tested for the ten classes of leaf. The details of each class are shown in table 5.1. This table also provides the information about the classes that are classified correctly.

The samples of the leaf taken are shown in figure 5.7 (a). The confusion matrix showing the information about the misclassification is shown in Figure 5.8 (a).

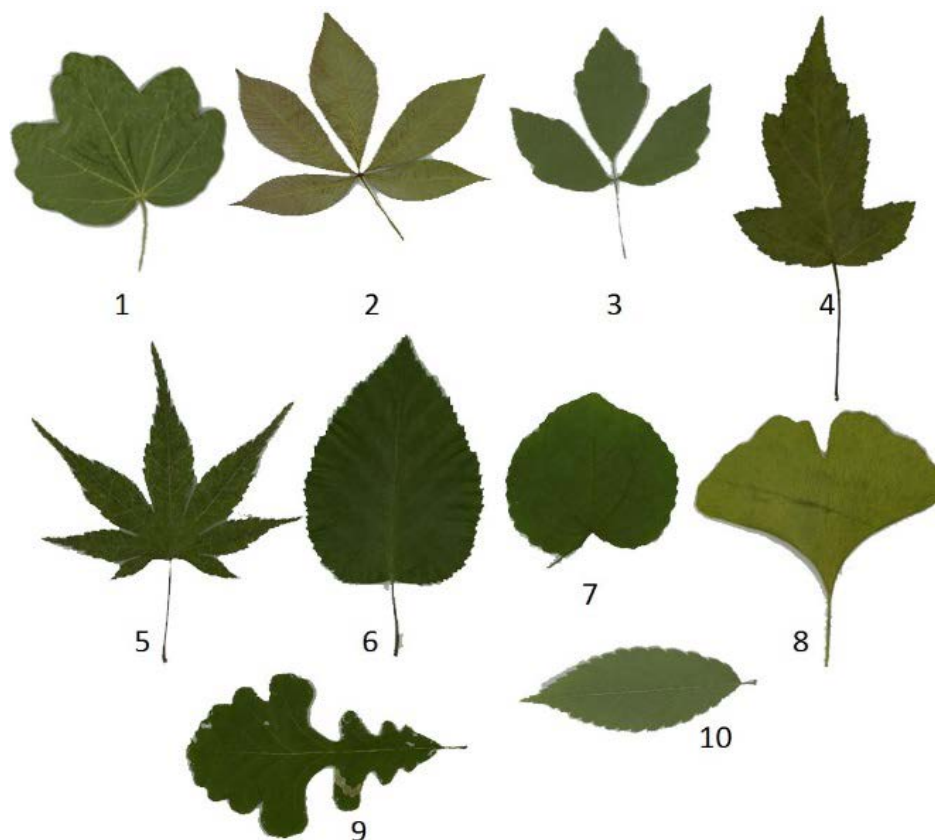


Figure 5.7 (a) Leaf Samples for each class of images for the names shown in table 5.1.

The method of HoCS (Histogram of Curved Scale) used in the application developed by *LeafSnap®* is also used in the above set of images. The confusion matrix for the above set of images is shown in figure 5.8 (b). On comparison it can be seen that the accuracy of the *HoCS* method is 86% which is less than the method used in this project.

**Confusion Matrix**

1	27	0	0	0	0	0	0	0	0	100%
	9.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
2	0	30	0	0	0	0	0	0	0	100%
	0.0%	10.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
3	0	0	30	0	0	0	0	0	0	100%
	0.0%	0.0%	10.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
4	0	0	0	30	0	0	0	0	0	100%
	0.0%	0.0%	0.0%	10.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5	0	1	0	0	18	1	0	0	0	90.0%
	0.0%	0.3%	0.0%	0.0%	6.2%	0.3%	0.0%	0.0%	0.0%	10.0%
6	0	0	0	0	1	32	0	0	0	97.0%
	0.0%	0.0%	0.0%	0.0%	0.3%	11.0%	0.0%	0.0%	0.0%	3.0%
7	0	0	0	0	0	1	29	0	0	96.7%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%	10.0%	0.0%	0.0%	3.3%
8	0	0	0	0	0	0	30	0	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.3%	0.0%	0.0%	0.0%
9	0	0	0	0	0	0	0	30	0	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.3%	0.0%	0.0%
10	0	0	0	0	0	0	0	0	30	100%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.3%	0.0%
	100%	96.8%	100%	100%	94.7%	94.1%	100%	100%	100%	98.6%
	0.0%	3.2%	0.0%	0.0%	5.3%	5.9%	0.0%	0.0%	0.0%	1.4%
	1	2	3	4	5	6	7	8	9	10
	<b>Target Class</b>									

**Output Class**

Figure 5.8 (a) Confusion Matrix for proposed method

1	22 7.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	1 0.3%	0 0.0%	3 1.0%	31.5% 18.5%
2	0 0.0%	23 7.9%	3 1.0%	1 0.3%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	1 0.3%	1 0.3%	76.7% 23.3%
3	0 0.0%	2 0.7%	24 8.3%	0 0.0%	1 0.3%	0 0.0%	0 0.0%	3 1.0%	0 0.0%	0 0.0%	80.0% 20.0%
4	0 0.0%	0 0.0%	1 0.3%	27 9.3%	2 0.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.0% 10.0%
5	1 0.3%	0 0.0%	0 0.0%	1 0.3%	16 5.5%	0 0.0%	0 0.0%	2 0.7%	0 0.0%	0 0.0%	90.0% 20.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	31 10.7%	2 0.7%	0 0.0%	0 0.0%	0 0.0%	93.9% 6.1%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.7%	28 9.7%	0 0.0%	0 0.0%	0 0.0%	93.3% 6.7%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.7%	0 0.0%	0 0.0%	28 9.7%	0 0.0%	0 0.0%	93.3% 6.7%
9	0 0.0%	1 0.3%	2 0.7%	1 0.3%	1 0.3%	0 0.0%	0 0.0%	1 0.3%	24 8.3%	0 0.0%	90.0% 20.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.7%	0 0.0%	0 0.0%	0 0.0%	28 9.7%	93.3% 6.7%
	95.7% 4.3%	88.5% 11.5%	80.0% 20.0%	90.0% 10.0%	72.7% 27.3%	83.8% 16.2%	93.3% 6.7%	80.0% 20.0%	96.0% 4.0%	87.5% 12.5%	86.6% 13.4%
	1	2	3	4	5	6	7	8	9	10	
	Target Class										
	Output Class										

Figure 5.8 (a) Confusion Matrix for HoCS method

## **Chapter 6: Conclusion & Future Scope**

### **Conclusion & Future Scope**

## 6.1 Conclusion

The experiments were conducted using various shape and texture descriptors. Various shape descriptors were used such as shape description using MPP (Minimum Perimeter Polygon), Shape Signatures, Shape contexts, Fourier Descriptors were used among which the Shape Context showed the most promising results. Identification of leaf was also done using the leaf tip and leaf bases, but it was observed that the matching was also not reliable because of the varying shapes of leaf tip and base for the same class of image. The shape alignment is also an image registration problem. The image alignment was at first done using Hough Transforms. But the image alignment for the leaves such as *Acer Plantanum* were not correctly aligned. Hence a more robust technique of alignment was done by estimating the parameters of the Affine Transform. This is a part of shape recognition using Shape Context hence the shape context produced the more promising results. A user interactive method for the alignment process is also implemented. Here the user marks roughly the major axis of plants, and then image is rotated in accordance with the axis.

Segmentation also played an important role in the classification. As a wrongly segmented image disrupts the entire flow of identification. The segmentation process by using graph cuts produces more efficient results.

From the very beginning, the image of sample has to be standardized before any operation is performed. The image is scaled, rotated and colors of image are enhanced at the pre-processing stage. The classification by MLP using backpropagation training proves to be an efficient way of classifying the images.

## 6.2 Future Scope

Though the results are promising yet there is always a way of improvement as no system is perfect. The efficiency of the system depends on the information supplied, which in turn depends on the number of features used in the classification of the system. Also for the improvement one need to constantly update the underlying technology.

Various features were used and tested in the project and several others can also be included. One important feature that can also be used in classification problem is using the leaf blade patterns on the leaf. This can be achieved by increasing the number of sampling points used in the

classification of the shapes using the shape context. This may create serious computational problems as processing a larger number of samples for each class of image increase the calculations required. Another way this information be included by using a shape descriptor that performs the computation based on the local information. Also to reduce the overall robustness of the system, better techniques of segmentation can be employed. As the current version of the Graph Cuts can be optimized. There are various papers that attempt to cut down the segmentation time are present. Also information such as color and the venation patterns can also increase the robustness of the system.

## References

- [1] Z.-Q. Zhao, L.-H. Ma, Y.-m. Cheung, X. Wu, Y. Tang and C. L. P. Chen, "Ap Leaf: An efficient android-based plant leaf identification system," *Neurocomputing*, vol. 151, no. 3, pp. 1112-1119, 2015.
- [2] C. Zhao, S. S. Chan, W.-K. Cham and L. M. Chu, "Plant identification using leaf shapes—A pattern counting approach," *Pattern Recognition*, vol. 48, no. 10, pp. 3203-3215, 2015.
- [3] A. K. Jain, A. Ross and S. Prabhakar, "An introduction to biometric recognition," *Circuits and Systems for Video Technology*, vol. I, no. 14, pp. 4-20, 2004.
- [4] L. Shen and L. Bai, "A review on Gabor wavelets for face recognition," *Pattern analysis and applications*, vol. 9, pp. 273-292, 2006.
- [5] J. Chaki and R. Parekh, "Plant Leaf Recognition using Gabor Filter," *International Journal of Computer Applications*, vol. 56, no. 10, pp. 26-29, 2012.
- [6] J. Chaki, R. Parekh and S. Bhattacharya, "Plant leaf recognition using texture and shape features with neural classifiers.," *Pattern Recognition Letters*, vol. 58, pp. 61-68, 2015.
- [7] S. Belongie, J. Malik and J. Puzich, "Shape context: A new descriptor for shape matching and object recognition," *NIPS*, vol. 2, pp. 3-10, 2000.
- [8] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions*, vol. 24, no. 7, pp. 971-987, 2002.
- [9] L. Nanni and A. Lumini, "Local binary patterns for a hybrid fingerprint matcher," *Pattern recognition*, vol. 41, no. 11, pp. 3461-3466, 2008.
- [10] T. Ahonen, A. Hadid and M. Pietikäinen, "Face recognition with local binary patterns," *Computer vision-eccv*, vol. Springer Berlin Heidelberg, pp. 469-481, 2004.



- [11] H. Ling and D. Jacobs, "Shape classification using the inner-distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 286-299, 2007.
- [12] N. Otsu, "A Threshold Selection Method from Gray-Level Histogram," *Automatica*, vol. 11, pp. 23-27, 1975.
- [13] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," *In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference*, vol. 1, pp. 105-112, 2001.
- [14] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley, 2014.
- [15] R. C. Gonzales, R. E. Woods and S. L. Eddins, *Digital Image Processing*, Chicago: Pearson Prentice Hall, 2004.
- [16] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, J. Kress, I. C. Lopez and J. V. B. Soares, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification," in *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, Italy, 2012.